

# Center for Empirically Based Software Engineering (CeBASE), Future Combat Systems, and Future Army Software Strategies



Dr. Vic Basili, Fraunhofer Center  
Dr. Barry Boehm, USC  
Dr. Richard Turner, OUSD(AT&L)



# CeBASE

- **Virtual Center for empirical software engineering**
  - NSF-initiated consortium of universities and research centers
  - Core researchers: academics with real world experience
    - Victor Basili - NASA, FC-MD/Industry, UMD
    - Barry Boehm - RAND, TRW, DARPA, USC
- **Strong track record of improving software productivity and quality**
  - **Invented software development and management approaches**
    - Spiral Model, Goal-Question-Metric, Theory-W, Experience Factory
  - **Created COCOMO family of estimation models**
    - Key to doubling TRW software productivity
  - **NASA Software Engineering Laboratory Experience Factory**
    - Decreased **Defect** rates by **75%** (87-91) **37%** (91-95)
    - Reduced **Cost** by **55%** (87-91) **42%** (91-95)
    - Improved **Reuse** by **300%** (87-91) **8%** (91-95)



# CeBASE and FCS

- **Member of FCS PM software support team with SEI and Sandia**
- **Membership on FCS Software Steering Committee with LSI, SEI, OSD**
- **Providing specific improvements in FCS software approach**
  - Elevating the level of software issues within the program
  - Building FCS software risk experience base
  - Stimulated rescoping of CTD demo program software
  - Identifying COTS integration strategies
  - Establishing cost and schedule estimates
  - Connecting LSI to DARPA IT programs
  - Performing software issue analysis as needed
  - Capturing experience to support later FCS block developments and future DoD acquisitions
- **Jointly funded by FCS PM and OUSD (AT&L) SIS Office**



# Significant FCS (and Future) Challenges

- **Unprecedented network-centric system**
  - FCS primary mission totally dependent on geographically-distributed integrated software environment
- **Disparate multi-stakeholder objectives**
  - Warfighters, logisticians, maintainers, strategists, acquirers, Congress...
- **Requisite software technology still being invented**
  - Scalable agent coordination, adaptive networks, security solutions, synchronized electronic upgrades, COTS integration, interoperability of independently evolving systems, -ility tradeoff analysis, ...
- **Rapid change in acquisition, technology and mission environments**
- **SoS software acquisition management**
  - What to contract for, tracking and synchronizing progresses, propagating and managing changes
  - How to capitalize on the experience from each block
- **SoS software interface definition and management**
  - Intra- and extra- System of Systems (SoS) interfaces critical success factor
  - SoS interfaces primarily facilitated or complicated by software



# Bottom Line

- **For complex, rapidly changing systems of systems, much can not be determined ahead of time by analysis**
- **Acquisition and development processes must be evolutionary**
  - Multiple spiral cycles of risk-driven specifying, prototyping, modeling, and simulation
  - Continuous measurement, feedback and learning from experience
  - Architectures that accommodate downstream needs
  - Continuous technology and mission-needs watch and change management
  - Milestone reviews that rigorously apply software success criteria
- **Software issues hide in nearly every program decision (high level and low) – need high-level visibility**
  - System architecture and design
  - Acquisition strategy
  - Dimensions of trade space
  - Interfaces, methodologies, ...



# Proposed Software Criteria for ASARC

- **A system built to the proposed life cycle software architecture will satisfy the following criteria**
  - Support the operational concept
  - Satisfy the requirements
  - Be faithful to the prototype(s)
  - Be buildable within the budgets and schedules in the plan
  - Show a viable business case
  - Establish key stakeholders' commitment to proceed
- **Criteria are Pass/Fail**
- **Supporting information documented in Feasibility Rationale**
- **All major risks resolved or covered by a risk management plan**
- **Reviewed by Independent Expert Program Review (IEPR) or equivalent**



# Back Up Slides



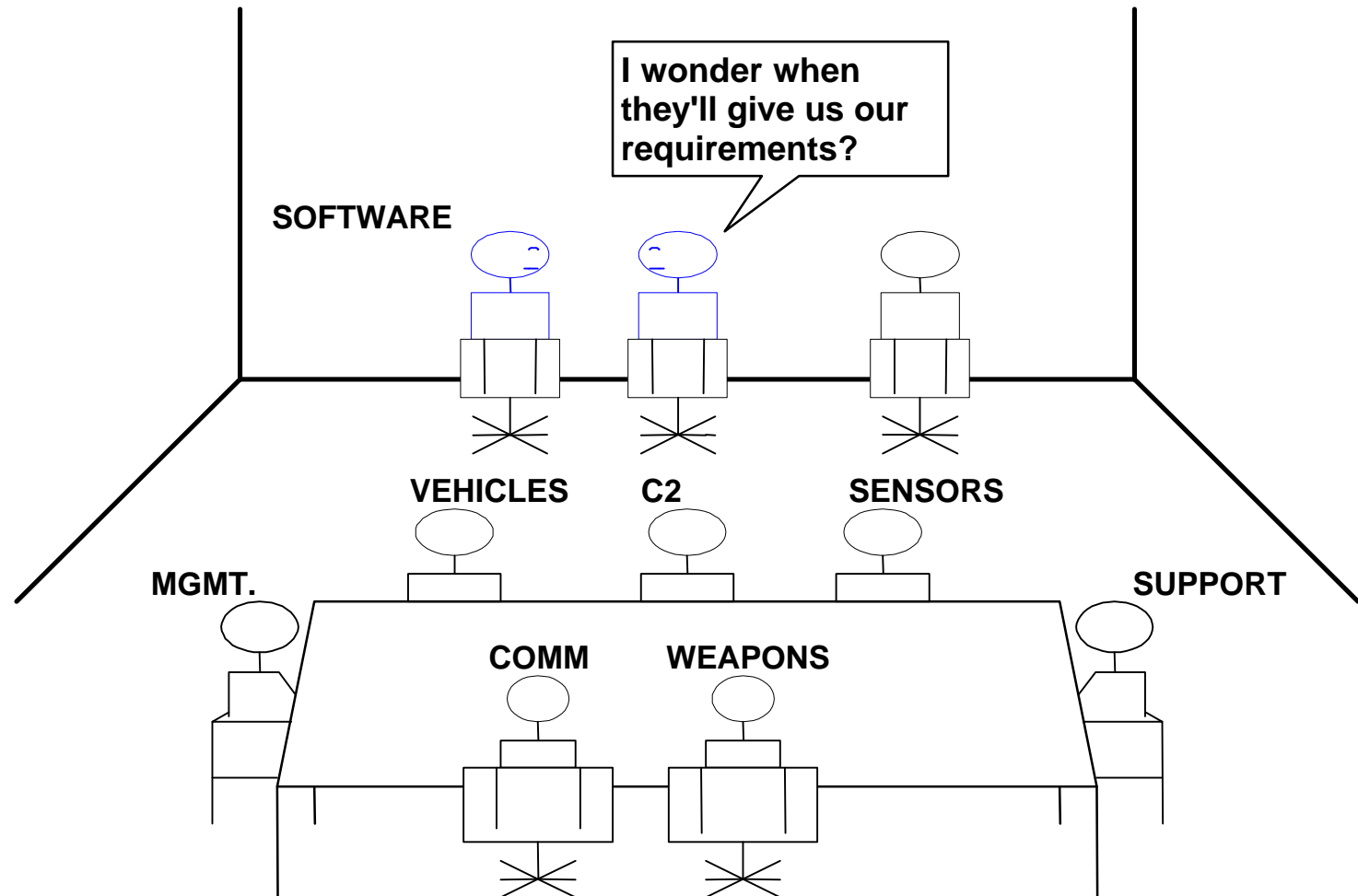
# How Software Differs from Hardware

- **Hard to visualize and kick the tires**
- **Easy to change if change directions anticipated; very hard otherwise**
- **Executed inflexibly by computers at speeds too fast for much human intervention**
- **Huge range of implementation options and decisions to orchestrate, understand, change-manage**
- **Increasingly commoditized around opaque commercial software packages**
- **Human talent to cope with these factors is very scarce; by far the most critical success factor**
- **Frequently gets onto system critical path**
- **Very easy to make SoS-level decisions with disastrous software consequences**
  - **Some of these come from believing current software myths**





# Software Expertise Needs to Be at the Center Table





# 7 Software Engineering Myths - 1

**Myth Number 1: COTS and commercial practices are the answer.**

*Fact: COTS works well in some situations but greatly increases risk in others. Commercial practices are optimized around rapidly bringing products to market, but with lower quality attribute levels than DoD mission critical systems require.*

**Myth Number 2. Commercial Industry will do DOD's needed software research.**

*Fact: Commercial industry does mass marketplace research.*

**Myth Number 3. The problem is software and programming methodology.**

*Fact: The problem is integrating software and system concerns.*

**Myth Number 4: SEI CMM for Software (or CMMI) is the answer.**

*Fact: Process maturity is only one aspect of software engineering*



## 7 Software Engineering Myths - 2

**Myth Number 5: Evolutionary Acquisition is the answer.**

*Fact: Evolutionary acquisition is a work in progress.*

**Myth Number 6: It's software—we can fix it later (add security, quality, other “-ilities”).**

*Fact: Most “-ilities” must be architected in, and can't be easily added later.*

**Myth Number 7: Create great components and the software engineering will take care of itself.**

*Fact: That's DoD's current course, and the problems aren't going away.*



# CeBASE Goals

- **Focus on high-risk software elements (COTS, subcontractor management, software size for CTD demos, moving-target requirements management)**
- **Build an evolving experience base of software risks, mitigation strategies, and lessons learned**
- **Establish and maintain a software cost/schedule estimation baseline**
- **Encourage a connection between LSI and science and technology sources**
- **Infuse Spiral Development concepts into the FCS Software Development Plan**
- **Shape the FCS software development plan and schedule for successful CTD and SDD performance**
- **Influence the ASARC/DAB success criteria to assure sound decisions**



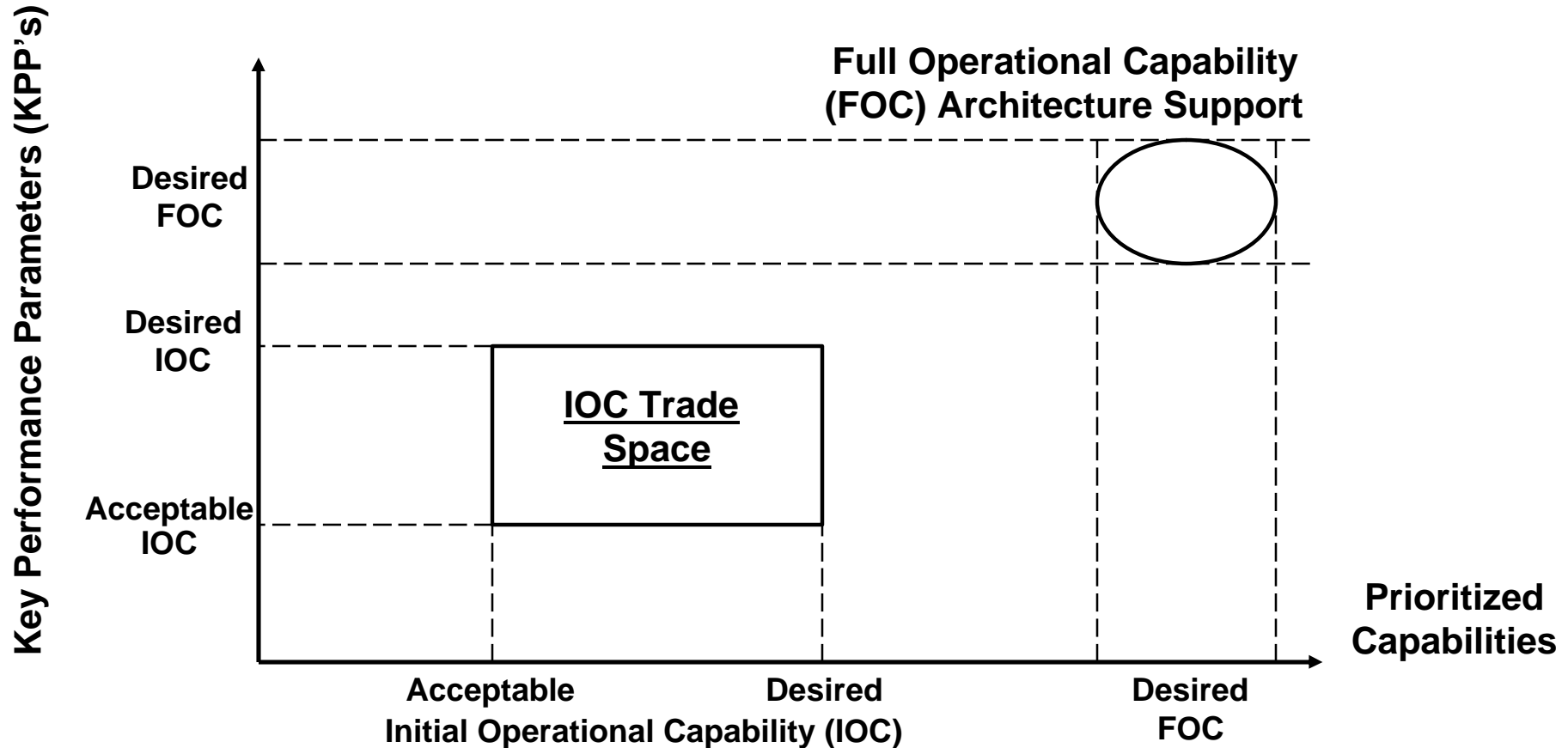
# The Way Ahead for FCS Software

- **Establish measurable goal ranges for all stakeholders**
  - FCS, DoD, TRADOC, LSI, CECOM, ...
- **Evaluate, iterate goal ranges based on M&S, architecture tradeoff and cost analysis**
  - Use with SAIV to define achievable block content
- **Capture experiences from each FCS block to**
  - Support development of subsequent blocks
  - Evaluate the effectiveness of current approaches
  - Track risk and identify effective mitigation strategies
  - Transfer what has been learned from FCS across other Army programs
- **Track technologies for use in future FCS blocks and Army programs**
  - Use FCS as a testbed framework for DARPA information technology initiatives
  - Encourages DARPA to focus on Army problems



# Avoid Point Solutions

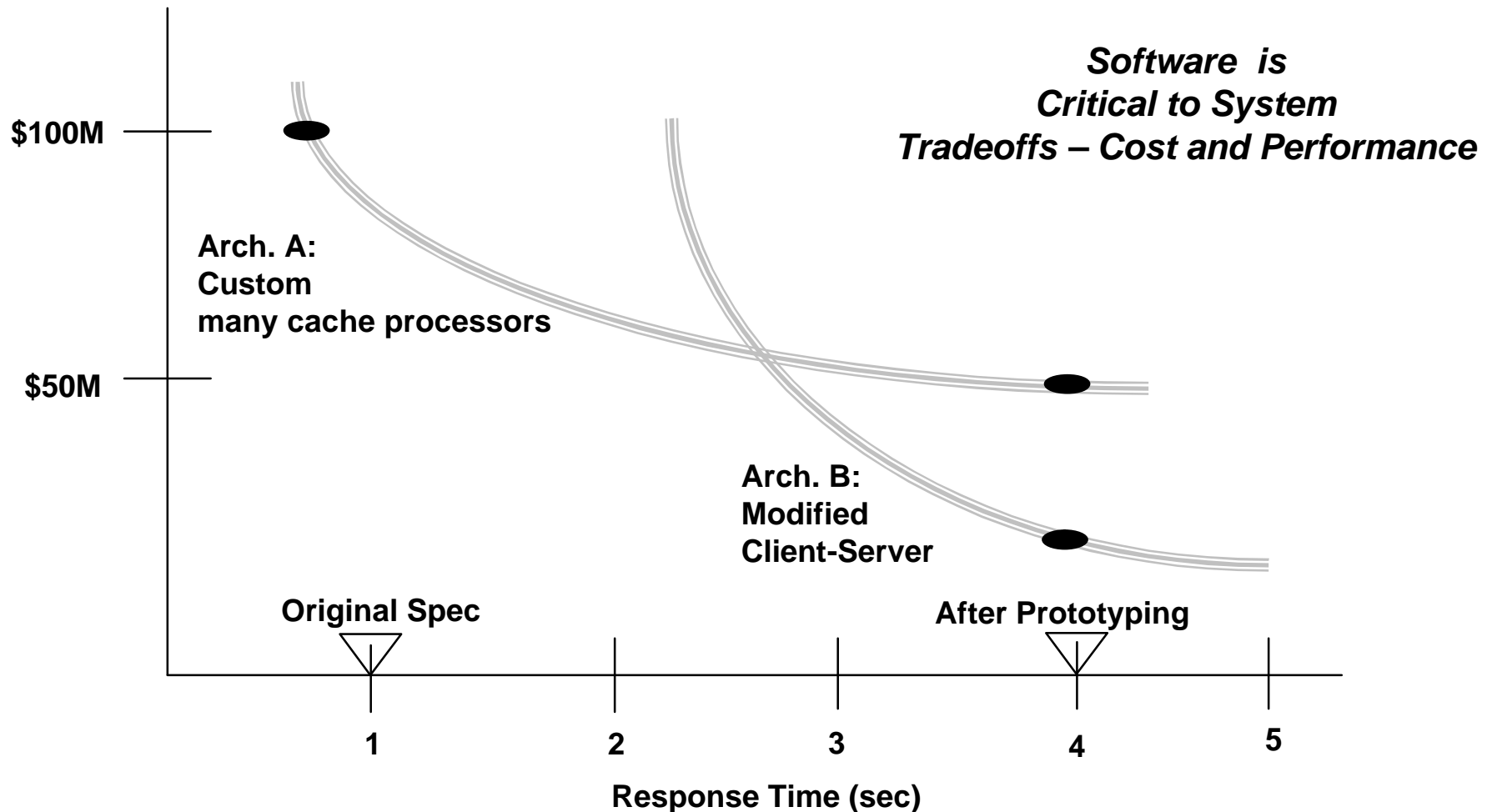
## Software Needed for Adaptability to Change



KPP's: cost, schedule, performance, dependability, interoperability, ...



# Example: Effect of system-imposed 1-second response time requirement





# Schedule As Independent Variable

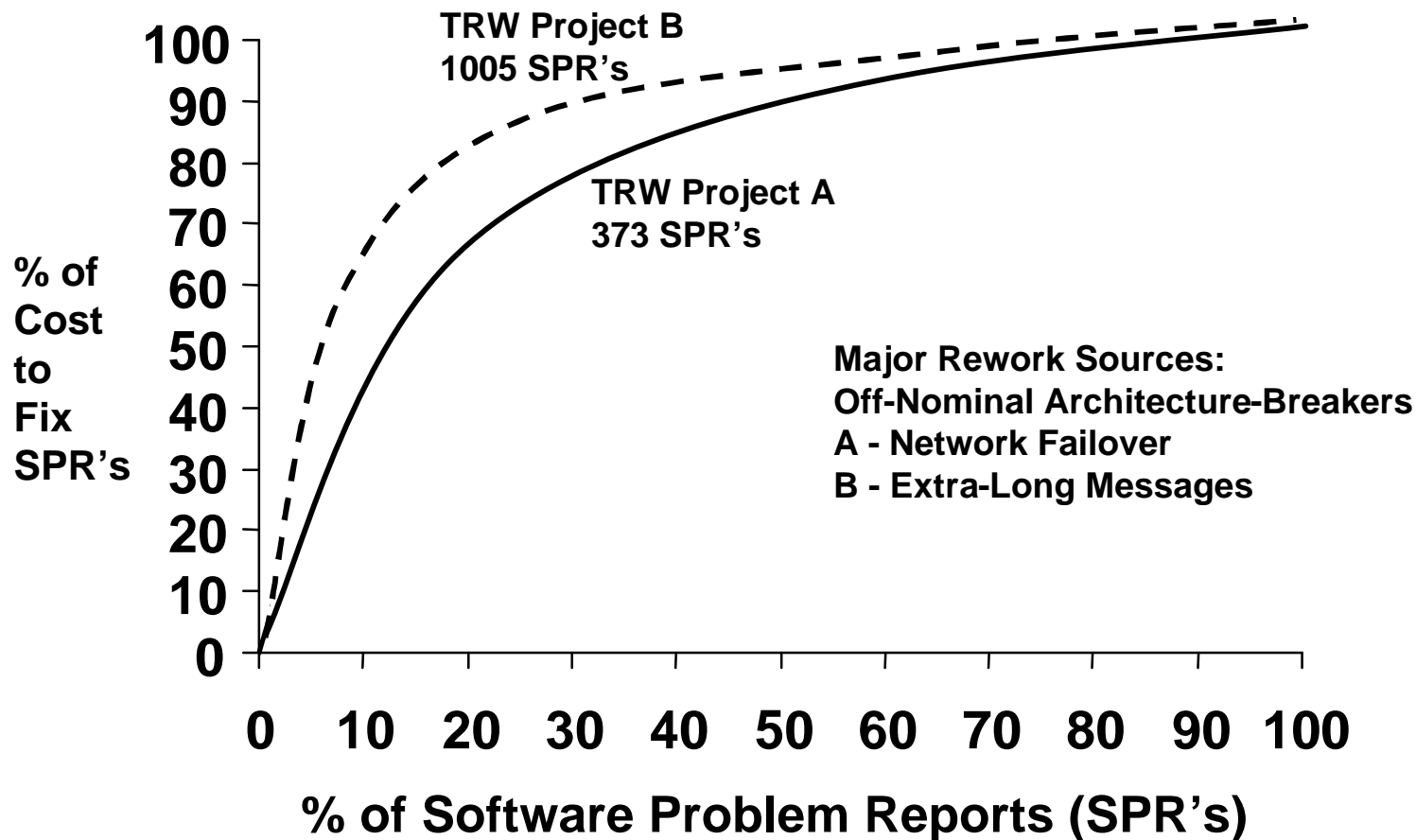
- **Features delivered adjusted to meet schedule**
- **SAIV involves**
  - Shared vision and expectations management
  - Feature prioritization
  - Schedule range estimation and core-capability determination
    - Top-priority features achievable within fixed schedule with 90% confidence
  - Architecting for ease of adding or dropping borderline-priority features
    - And for accommodating post-IOC directions of growth
  - Incremental development
    - Core capability as increment 1
  - Change and progress monitoring and control
    - Add or drop borderline-priority features to meet schedule





# Example: Effects of Point Solution IOC

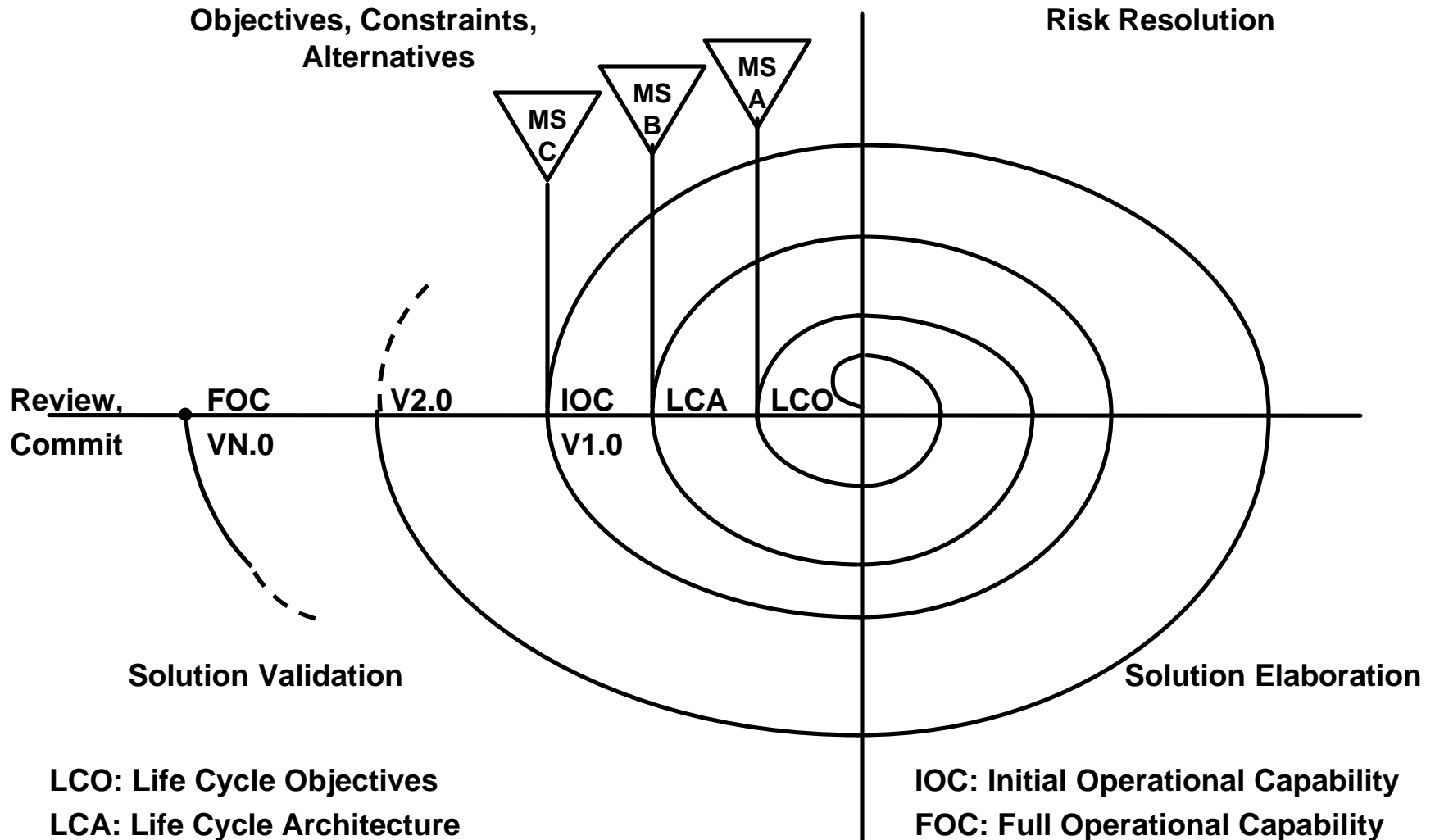
## -Pareto Analysis of Rework Costs





# The CeBASE Spiral and DoD 5000.2

– Cross Talk, May 2001

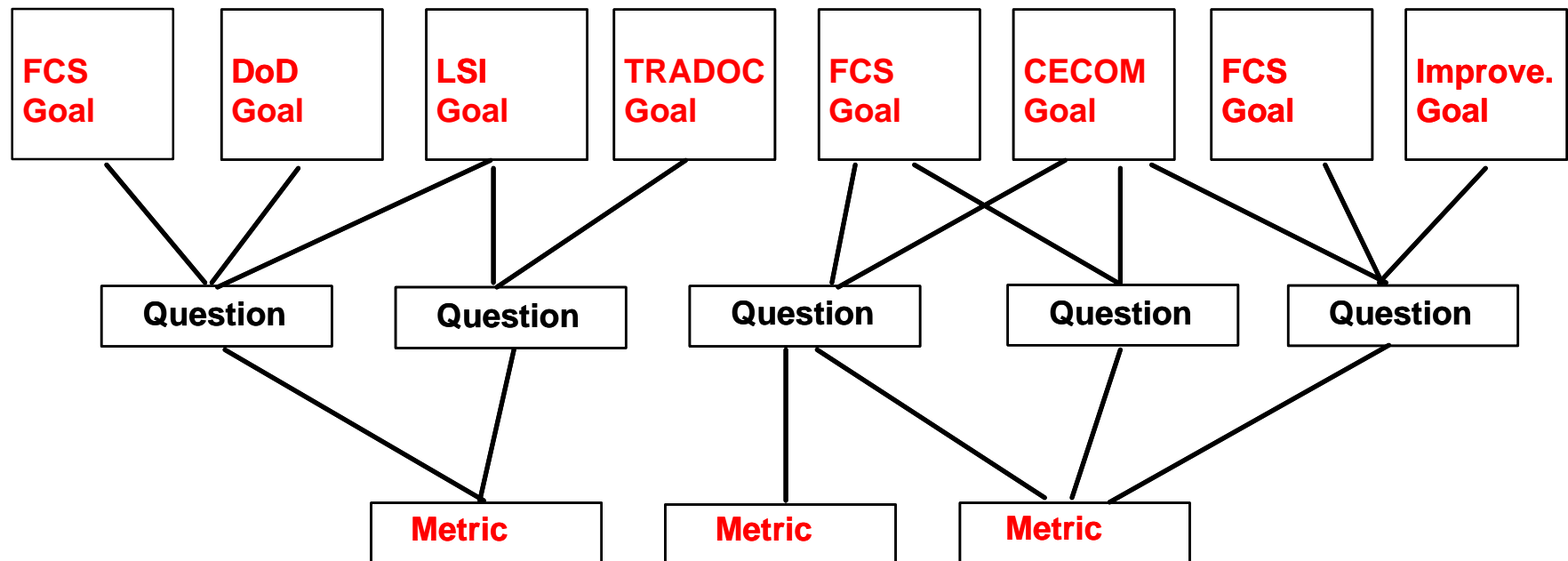




# Using the Goal/Question/Metric Paradigm to Tie FCS Goals to Measurement

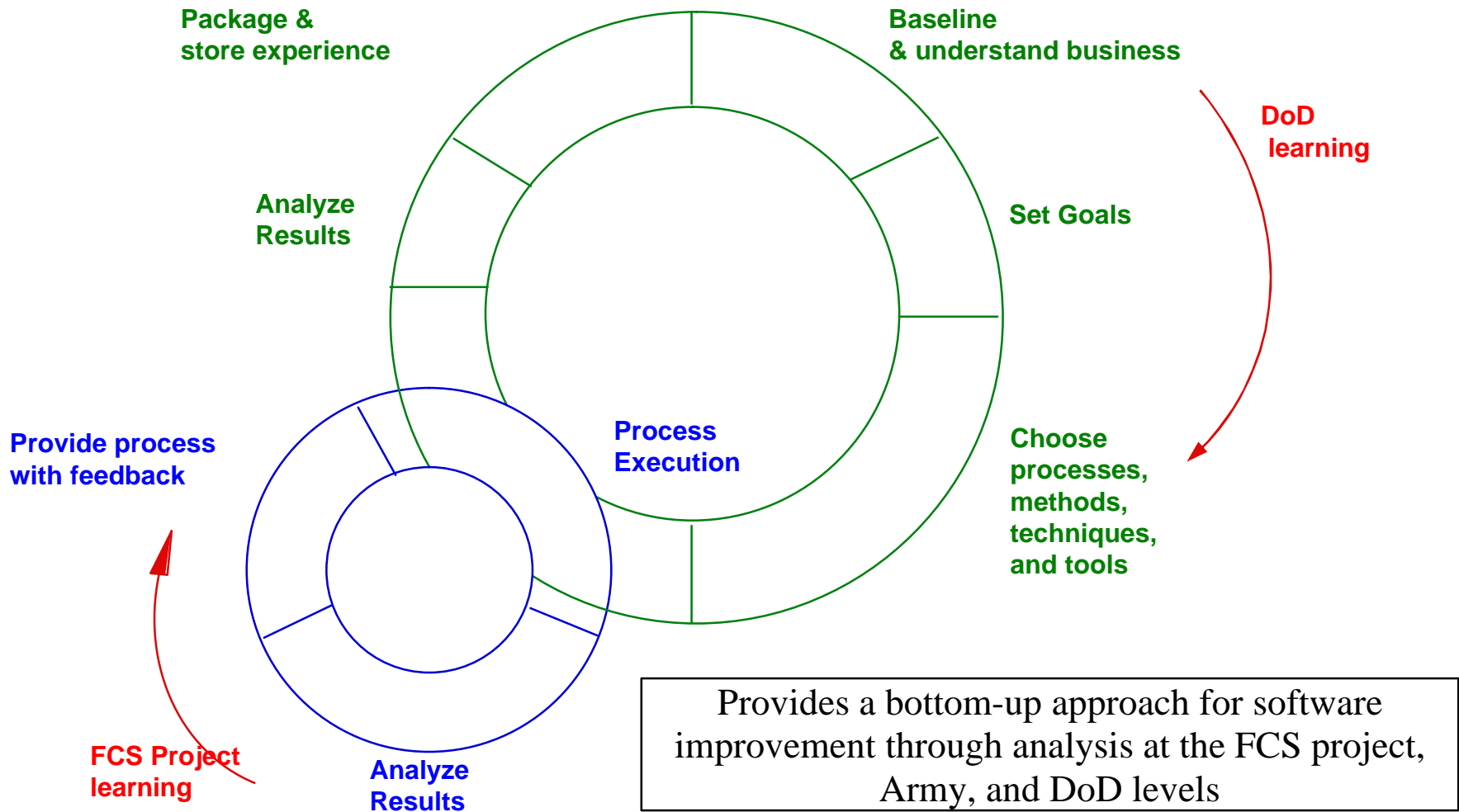
## Measurement Infrastructure

Each metric supports multiple DoD stakeholder goals while the questions help focus metric selection and in-process analysis





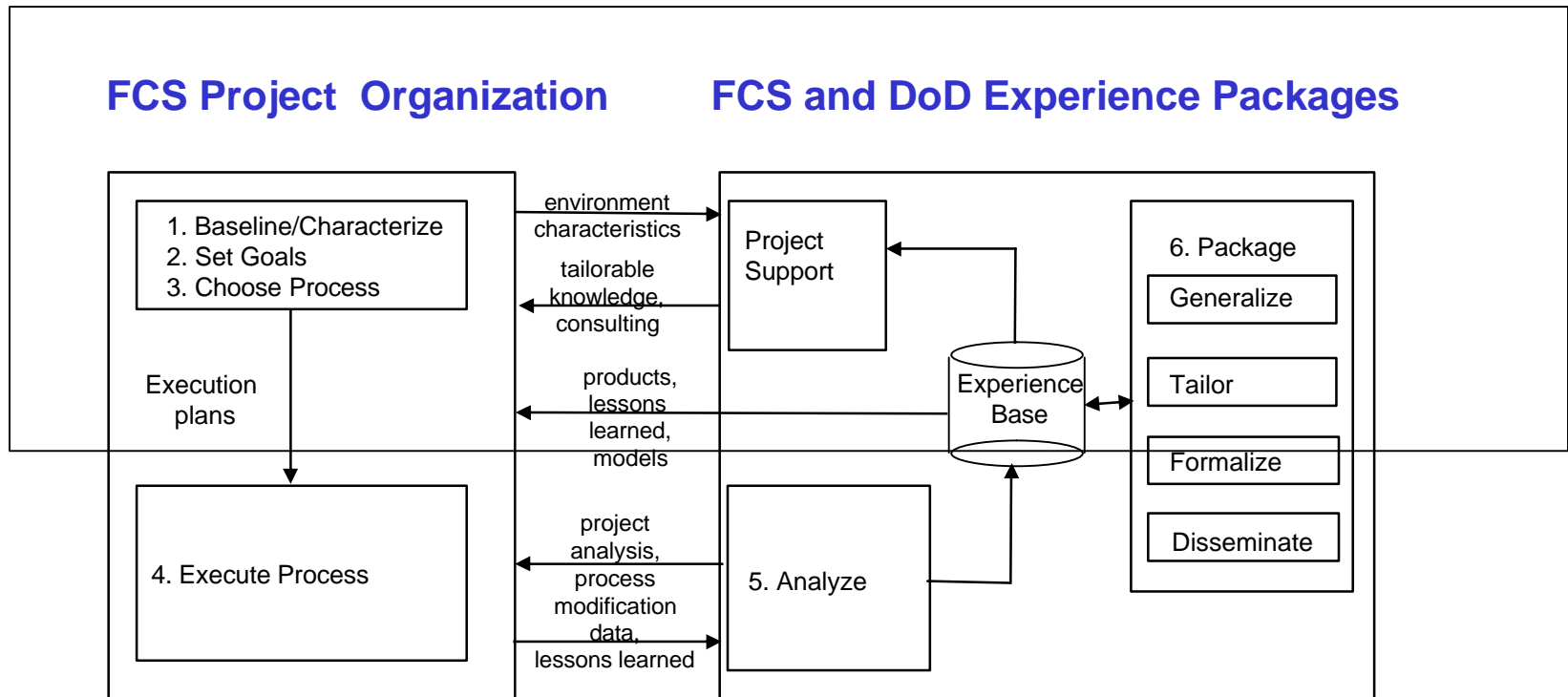
# Quality Improvement Paradigm





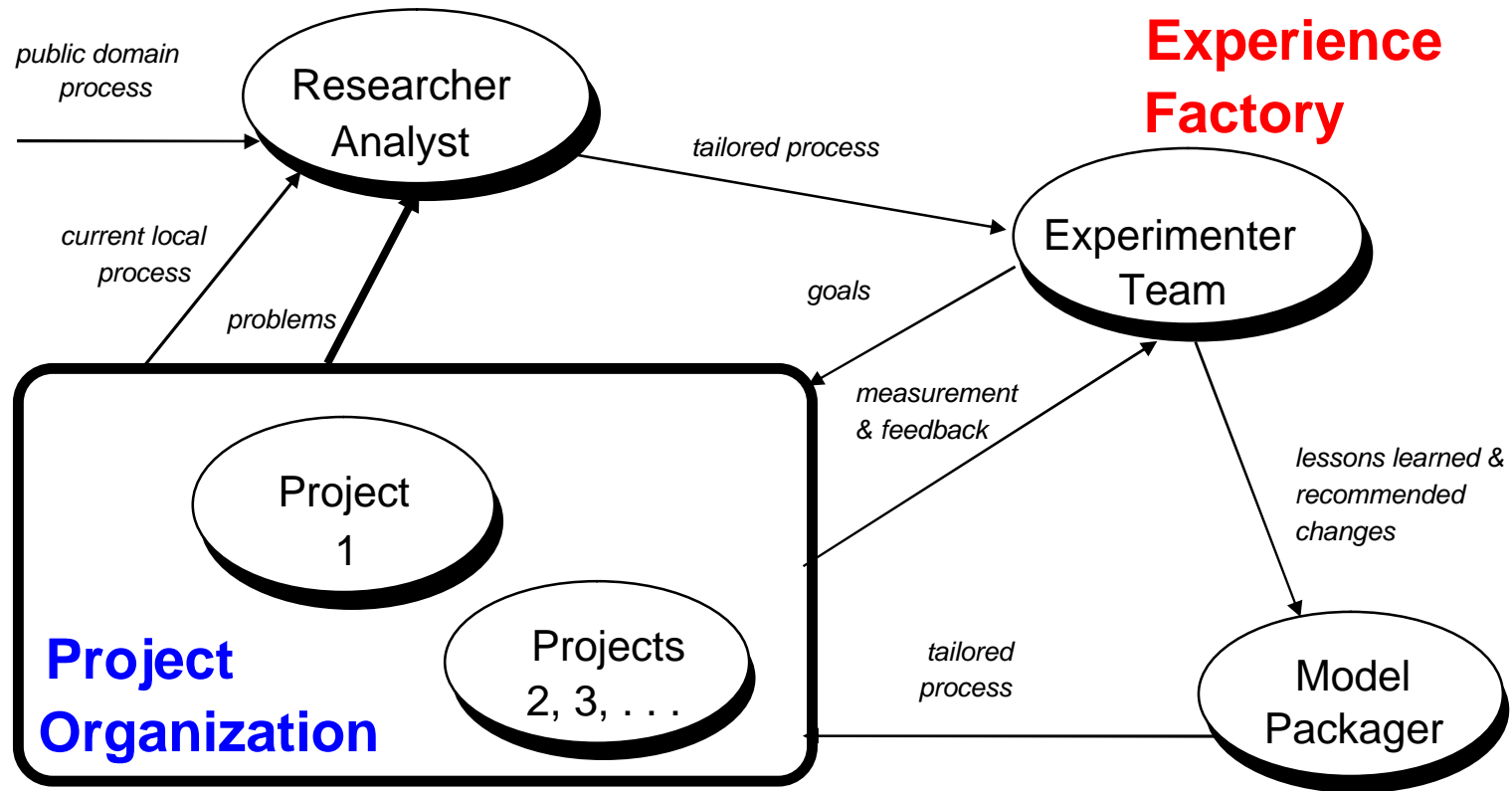
# Experience Factory Organization

Capture experience from the unprecedented FCS program to support future FCS activities and DoD programs by accumulating knowledge and analyzing it continually to help meet near- and long-term goals





# Experience Factory Elements



FCS experience can be captured from and spread across multiple programs



# NASA Goddard Experience: An Experience Factory Example

## Continuous Improvement in Ground Support Systems

Decreased **Development Defect rates** by  
**75%** (87 - 91)    **37%** (91 - 95)

Reduced **Cost** by  
**55%** (87 - 91)    **42%** (91 - 95)

Improved **Reuse** by  
**300%** (87 - 91)    **8%** (91 - 95)

Increased **Functionality** five-fold (76 - 92)



# Experience Factory Framework - I

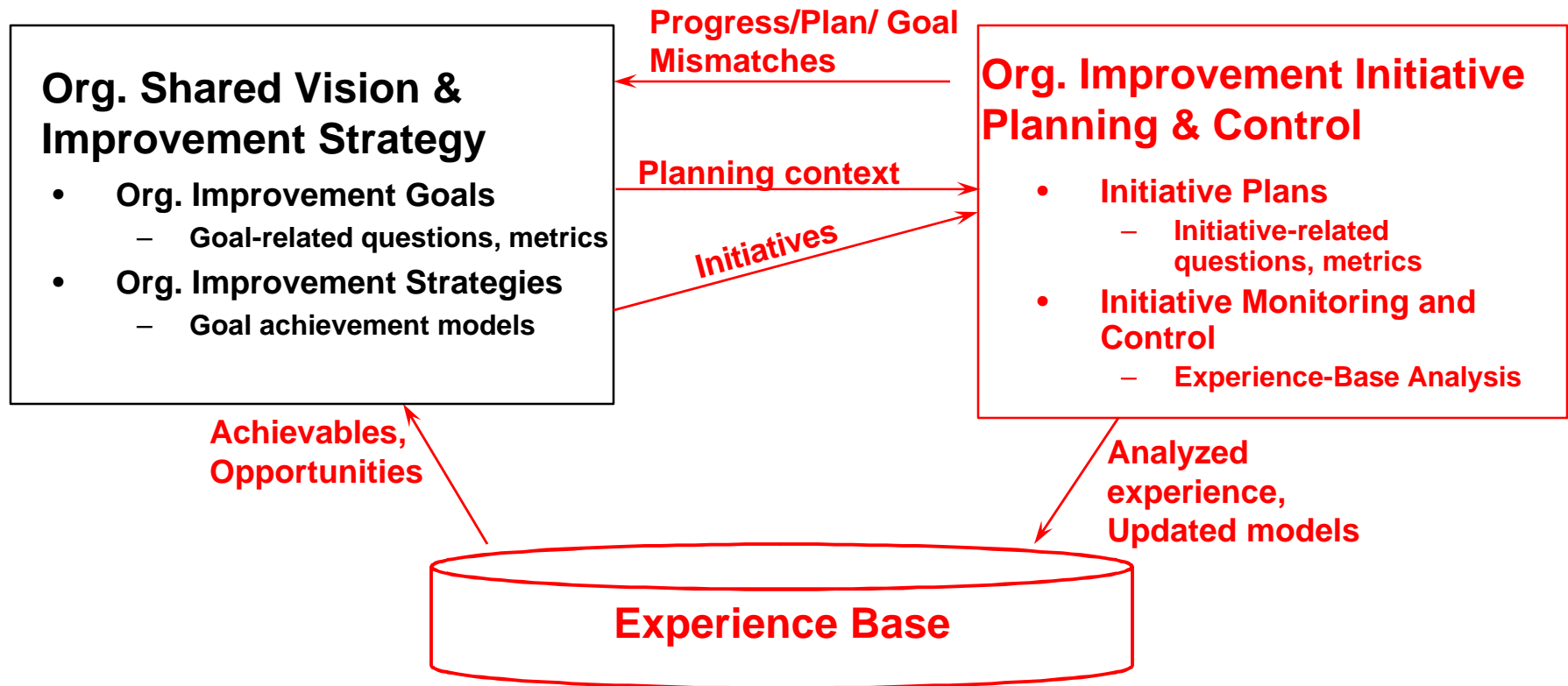
## **Org. Shared Vision & Improvement Strategy**

- **Org. Improvement Goals**
  - Goal-related questions, metrics
- **Org. Improvement Strategies**
  - Goal achievement models



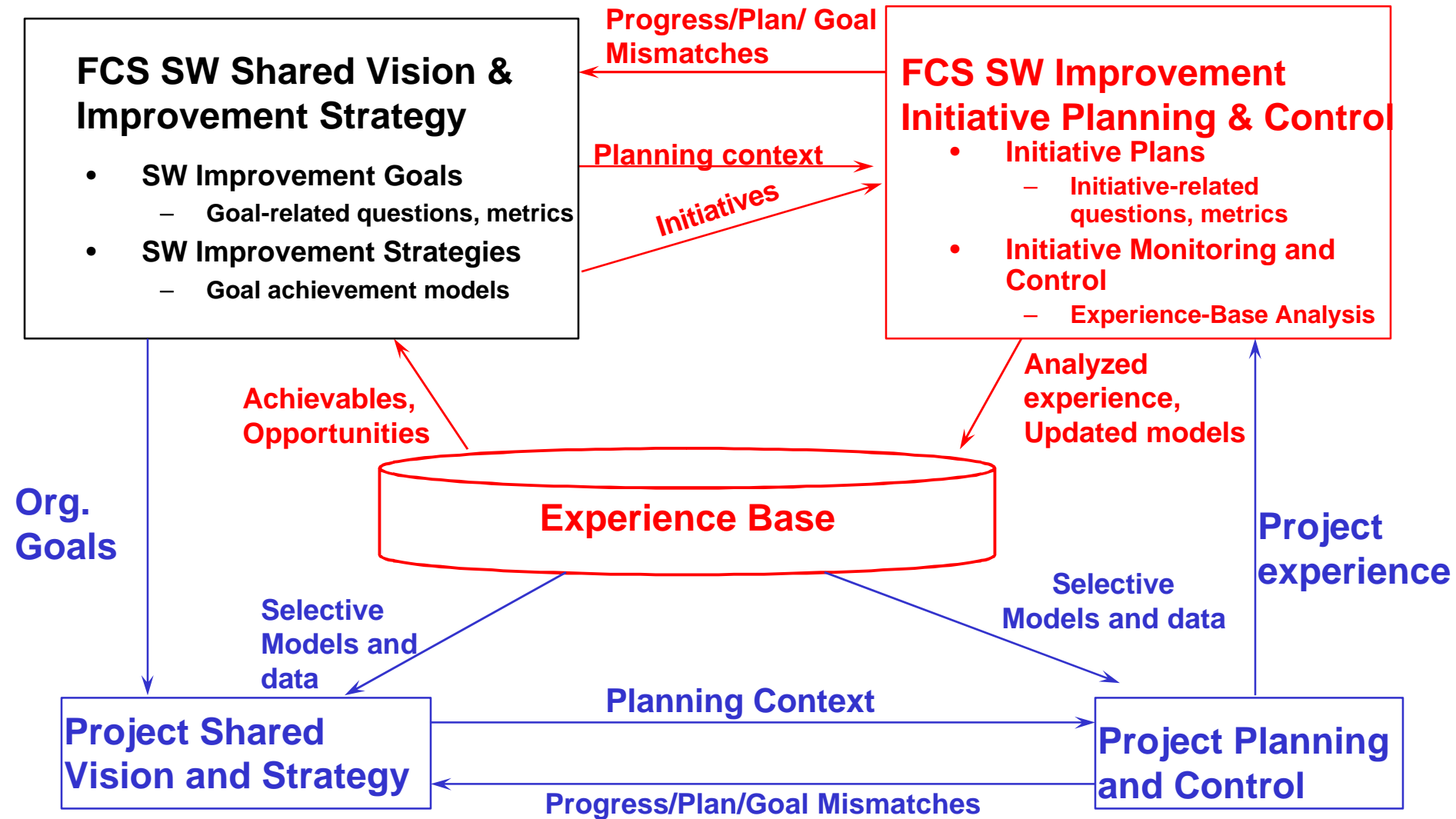


# Experience Factory Framework - II





# Experience Factory Framework - FCS





# **MBASE Shared Vision Package**

## **-Section 2 of Operational Capability Description**

### **2. Shared Vision**

#### **2.1 System Capability Description**

##### **2.1.1 Benefits Realized**

##### **2.1.2 Results Chain**

#### **2.2 Key Stakeholders**

- Roles, responsibilities, contributions to Results Chain

#### **2.3 System Boundary and Environment**

- Context Diagram

#### **2.4 Major Project Constraints**

##### **Add for Scaling Up**

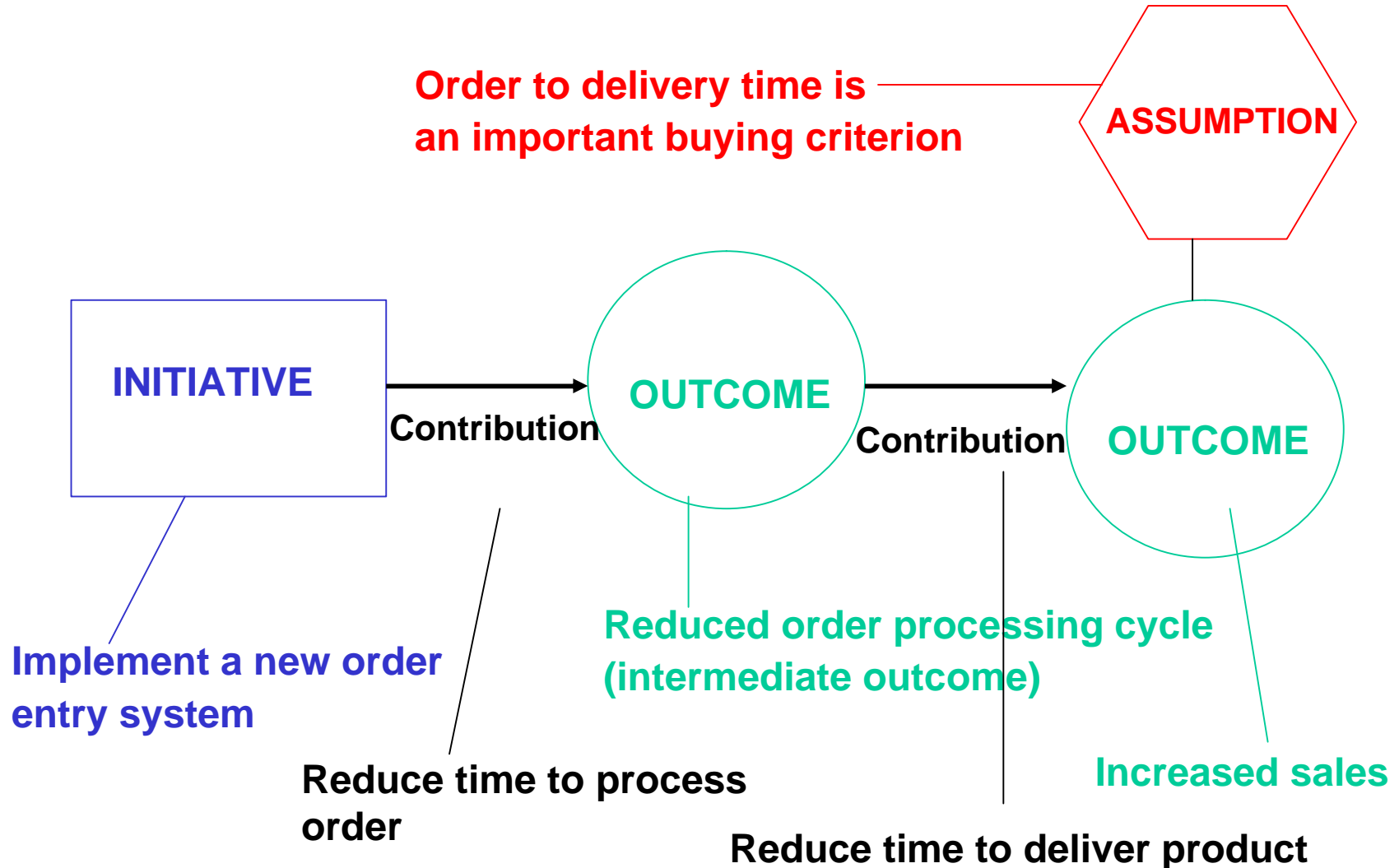
##### **2.5 Top-level business case**

##### **2.6 Inception phase plan, resources required**

##### **2.7 Initial Spiral objectives, constraints, alternatives, risks**



# DMR/BRA Results Chain





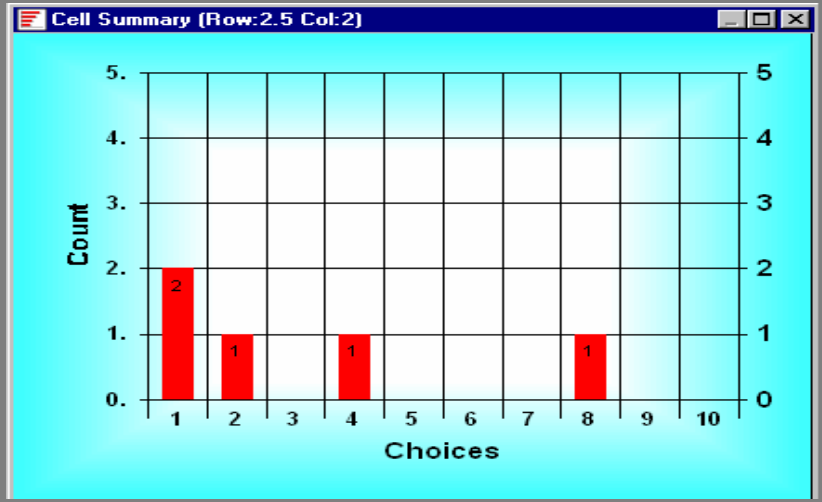
## EasyWinWin Activities

***Now a commercial  
product from  
GroupSystems.com***

<i>Activity</i>	<i>"ThinkLet"</i>	<i>Tool</i>
Elaborate domain taxonomy	Could-be/ Should-be	GroupOutliner
Brainstorm stakeholder interests	Free Brainstorming	Electronic Brainstorming
Converge on win conditions	FastFocus	Categorizer
Capture domain language	TermCapture	Topic Commenter
Prioritize win conditions	MultiCriteria	Alternative Analysis
Elaborate Conflicts, Constraints, Issues	CrowBar, MultiPass	GroupOutliner
Elaborate Options	MultiPass	GroupOutliner
Negotiate Agreements	MultiPass	GroupOutliner

**Red cells indicate lack of consensus.**

**Oral discussion of cell graph reveals unshared information, unnoticed assumptions, hidden issues, constraints, etc.**



	Features	Importance	Use of Implementation	Total	Mean
2.	Application Capabilities				
2.1	W2 Integrate banner ads with email and chat	8.00	6.50	16.50	8.25
2.2	W3 The banner will provide a link to the universit	10.00	10.00	20.00	10.00
2.3	W4 Interface for advertisers to select their sched	8.67	3.00	11.67	5.83
2.4	W5 Default banner of bookstore if no other events	8.00	10.00	18.00	9.00
2.5	W6 The site management must have a website which	10.00	10.00	20.00	10.00
2.6	W7 Different kinds of advertising, including sales	10.00	10.00	20.00	10.00
2.7	W8 Flexible text on banners	10.00	5.00	15.00	7.50
2.8	W9 Display address of the bookstore, a map of it a	4.00	7.50	11.50	5.75
2.9	W10 Ads must be hyperlinked so that users can clic	7.33	6.00	13.33	6.67
2.10	W11 Link to bookstore site (incl book's prices)	9.33	10.00	19.33	9.67
2.11	W12 Web statistics tracking to determine number of	8.00	4.00	12.00	6.00
2.12	W13 Input of banner contents to admin via email	5.50	10.00	15.50	7.75